

---

## Mètode de la classe cua per a moure el segon element al finalX57820\_ca

---

Implementeu un nou mètode de la classe Queue que tregui l'element situat en segona posició, i el posi al final de la cua. En altres paraules, si el contingut de la cua és  $[a_1, a_2, a_3, \dots, a_n]$ , a on els elements es presenten des del front de la cua fins al final, llavors recol·locar el segon element de la cua al final d'aquesta la deixa així:  $[a_1, a_3, \dots, a_n, a_2]$ . El els casos especials en que la cua té 0 o 1 o 2 elements, el mètode no fa res.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `queue.old.hpp`, a on hi ha una implementació de la classe genèrica Queue. En primer lloc, haureu de fer:

```
cp queue.old.hpp queue.hpp
```

A continuació, haureu de buscar dins `queue.hpp` la part:

```
// Pre:
// Post: El segon element de la cua s'ha mogut a l'última posició.
//       Si hi havia menys de tres elements a la cua, llavors res ha canviat.
// Descomenteu les següents dues línies i implementeu la funció:
// void moveSecondToLast() {
// }
```

Haureu de descomentar les dues línies que s'indiquen i implementar aquest mètode. No toqueu la resta de la implementació de la classe, excepte si, per algun motiu, considereu que necessiteu afegir algun mètode auxiliar a la part privada.

Preferiblement, haurieu d'aconseguir implementar el mètode a base d'intercanviar els punters de l'objecte. De fet, una implementació que no es basi en això potser us permetrà passar els jocs de proves públics (i així obtenir una part de la nota), però possiblement no els privats, doncs serà massa lenta.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `program.cpp` (programa principal) i `Makefile` per a compilar. Per a pujar la vostra solució, heu de crear el fitxer `solution.tar` així:

```
tar cf solution.tar queue.hpp
```

### Entrada

La entrada del programa és una seqüència d'instruccions del següent tipus que s'aniran aplicant sobre una cua de strings que se suposa inicialment buida:

```
push x (x és un string)
pop
front
size
moveSecondToLast
```

Se suposa que la seqüència d'entrada serà correcta (sense pop ni front sobre cua buida).

El programa principal que us oferim ja s'encarrega de llegir aquestes entrades i fer les crides als corresponents mètodes de la classe cua. Només cal que implementeu el mètode abans esmentat.

## Sortida

Per a cada instrucció `front`, s'escriurà el front actual de la cua. Per a cada instrucció `size`, s'escriurà la mida de la cua. El programa que us oferim ja fa això. Només cal que implementeu el mètode abans esmentat.

### Exemple d'entrada 1

```
size
moveSecondToLast
size
push a
front
size
moveSecondToLast
push b
front
size
moveSecondToLast
push c
front
size
moveSecondToLast
pop
front
size
moveSecondToLast
push d
push e
push f
size
moveSecondToLast
moveSecondToLast
moveSecondToLast
size
front
pop
front
pop
front
pop
front
pop
front
pop
size
```

### Exemple d'entrada 2

```
push l
push bm
pop
front
size
moveSecondToLast
moveSecondToLast
push ar
push w
push y
push d
```

### Exemple de sortida 1

```
0
0
a
1
a
2
a
3
c
2
5
5
c
f
b
d
e
0
```

```
moveSecondToLast
push c
push rj
push w
moveSecondToLast
push sj
push ld
push f
push r
push yn
pop
front
size
```

```
pop
front
size
moveSecondToLast
push g
moveSecondToLast
moveSecondToLast
moveSecondToLast
pop
front
size
push r
push ln
pop
front
size
push p
push wk
push p
pop
front
size
pop
front
size
```

## Exemple de sortida 2

```
bm
1
Y
12
d
11
w
11
sj
12
ld
14
f
13
```

## Observació

Afegiu comentaris explicatius si aquests són convenientes per tal d'entendre el vostre codi.  
Avaluació sobre 10 punts:

- Solució lenta: 6 punts.
- solució ràpida: 10 punts.

Entenem com a solució lenta una que és correcta i capaç de superar els jocs de proves públics.  
Entenem com a solució ràpida una que és correcta i capaç de superar els jocs de proves públics i privats.

## Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:17:10.913Z

© *Jutge.org*, 2006–2026.  
<https://jutge.org>